A note on the multiple unicast capacity of directed acyclic networks

Shurui Huang, *Student Member, IEEE* and Aditya Ramamoorthy, *Member, IEEE*
Department of Electrical and Computer Engineering
Iowa State University, Ames, Iowa 50011
Email: {hshurui, adityar}@iastate.edu

***Abstract*—We consider the multiple unicast problem under network coding over directed acyclic networks with unit capacity edges. There is a set of n source-terminal ($s_i - t_i$) pairs that wish to communicate at unit rate over this network. The connectivity between the $s_i - t_i$ pairs is quantified by means of a connectivity level vector, $[k_1 \ k_2 \dots \ k_n]$ such that there exist k_i edge-disjoint paths between s_i and t_i. Our main aim is to characterize the feasibility of achieving this for different values of n and $[k_1 \ \dots \ k_n]$. For 3 unicast connections ($n = 3$), we characterize several achievable and unachievable values of the connectivity 3-tuple. In addition, in this work, we have found certain network topologies, and capacity characterizations that are useful in understanding the case of general n.**

I. Introduction

Network coding has emerged as an interesting alternative to routing in the next generation of networks. In particular, it is well-known that the network coding is a provably capacity achieving strategy for network multicast. The work of [1] provides a nice algebraic framework for reasoning about network coding, and significantly simplifies the proofs of [2], and suggests network code design schemes. However, general network connections, such as multiple unicasts are more difficult to understand under network coding. In a multiple unicast connection, there are several source terminal pairs; each source wishes to communicate to its corresponding terminal. The goal is to find a characterization of the network resources required to support this connection using network coding.

The multiple unicast problem has been examined for both directed acyclic networks [3][4][5] and undirected networks [6] in previous work. The work of [7], provides an information theoretic characterization for directed acyclic networks. However, in practice, evaluating these bounds becomes computationally infeasible even for small networks because of the large number of inequalities that are involved. Moreover, these approaches do not suggest any constructive code design approaches. The work of [4], considers the multiple unicast problem in the case of two source-terminal pairs, while the work of [3] attempts to address it by packing butterfly networks within the original graph. Das et al. [8] consider the multiple unicast problem with an interference alignment approach. For undirected networks, there is open conjecture as to whether there is any advantage to using network coding as compared to routing ([6]). Multiple unicast in the presence of link faults and errors, under certain restricted (though realistic) network topologies has been studied in [9][10].

In this work our aim is to better understand the combinatorial aspects of the multiple unicast problem over directed acyclic networks. We consider a network G, with unit capacity edges and source-terminal pairs, $s_i - t_i, i = 1, \dots, n$, such that the maximum flow from s_i to t_i is k_i. Each source contains a unit-entropy message that needs to be communicated to the corresponding terminal. Our objective is to determine whether there exist feasible network codes that can satisfy the demands of the terminals. This is motivated by a need to find characterizations that can be determined in a computationally efficient manner.

A. Main Contributions

- For the case of three unicast sessions ($n = 3$), we identify all feasible and infeasible connectivity levels $[k_1 \ k_2 \ k_3]$. For the feasible cases, we provide efficient linear network code assignments. For the infeasible cases, we provide counter-examples, i.e., instances of graphs where the multiple unicast cannot be supported under any (potentially nonlinear) network coding scheme.
- We identify certain feasible/infeasible instances with two unicast sessions, where the message entropies are different. These are used to arrive at conclusions for the problem in the case of higher n (> 3).

This paper is organized as follows. In section II, we introduce several concepts that will be used throughout the paper. We also describe the precise problem formulation. Section III identifies the feasible routing connectivity levels. We discuss the network coding case in Section IV. Counter examples are given for infeasible connectivity levels. A feasible connectivity level with vector network coding solution is also provided. Section V concludes the paper.

II. Preliminaries

We represent the network as a directed acyclic graph $G = (V, E)$. Each edge $e \in E$ has unit capacity and can transmit one symbol from a finite field of size q per unit time (we are free to choose q large enough). If a given edge has higher capacity, it can be treated as multiple unit capacity edges. A directed edge e between nodes i and j is represented as (i, j), so that $head(e) = j$ and $tail(e) = i$. A path between two nodes i and j is a sequence of edges $\{e_1, e_2, \dots, e_k\}$ such that

This research was supported in part by NSF grant CCF-1018148.

$tail(e_1) = i, head(e_k) = j$ and $head(e_i) = tail(e_{i+1}), i = 1, \ldots, k-1$. The network contains a set of n source nodes s_i and n terminal nodes $t_i, i = 1, \ldots n$. Each source node s_i observes a discrete integer-entropy source, that needs to be communicated to terminal t_i. Without loss of generality, we assume that the source (terminal) nodes do not have incoming (outgoing) edges. If this is not the case one can always introduce an artificial source (terminal) node connected to the original source (terminal) node by an edge of sufficiently large capacity that has no incoming (outgoing) edges.

We now discuss the network coding model under consideration in this paper. For the sake of simplicity, suppose that each source has unit-entropy, denoted by X_i. In scalar linear network coding, the signal on an edge (i, j), is a linear combination of the signals on the incoming edges on i or the source signals at i (if i is a source). We shall only be concerned with networks that are directed acyclic and can therefore be treated as delay-free networks [1]. Let Y_{e_i} (such that $tail(e_i) = k$ and $head(e_i) = l$) denote the signal on edge $e_i \in E$. Then, we have

$$Y_{e_i} = \sum_{\{e_j | head(e_j)=k\}} f_{j,i} Y_{e_j} \text{ if } k \in V \backslash \{s_1, \ldots, s_n\}, \text{ and}$$

$$Y_{e_i} = \sum_{j=1}^{n} a_{j,i} X_j \quad \text{where } a_{j,i} = 0 \text{ if } X_j \text{ is not observed at } k.$$

The coefficients $a_{j,i}$ and $f_{j,i}$ are from the operational field. Note that since the graph is directed acyclic, it is equivalently possible to express Y_{e_i} for an edge e_i in terms of the sources X_j's. If $Y_{e_i} = \sum_{k=1}^{n} \beta_{e_i,k} X_k$ then we say that the global coding vector of edge e_i is $\boldsymbol{\beta}_{e_i} = [\beta_{e_i,1} \ \cdots \ \beta_{e_i,n}]$. We shall also occasionally use the term coding vector instead of global coding vector in this paper. We say that a node i (or edge e_i) is downstream of another node j (or edge e_j) if there exists a path from j (or e_j) to i (or e_i).

Vector linear network coding is a generalization of the scalar case, where we code across the source symbols in time, and the intermediate nodes can implement more powerful operations. Formally, suppose that the network is used over T time units. We treat this case as follows. Source node s_i now observes a vector source $[X_i^{(1)} \ \ldots \ X_i^{(T)}]$. Each edge in the original graph is replaced by T parallel edges. In this graph, suppose that a node j has a set of β_{inc} incoming edges over which it receives a certain number of symbols, and β_{out} outgoing edges. Under vector network coding, j chooses a matrix of dimension $\beta_{out} \times \beta_{inc}$. Each row of this matrix corresponds to the local coding vector of an outgoing edge from j.

Note that the general multiple unicast problem, where edges have different capacities and the sources have different entropies can be cast in the above framework by splitting higher capacity edges into parallel unit capacity edges, a higher entropy source into multiple, collocated unit-entropy sources; and the corresponding terminal node into multiple, collocated terminal nodes.

An instance of the multiple unicast problem is specified by the graph G and the source terminal pairs $s_i - t_i, i = 1, \ldots, n$, and is denoted $< G, \{s_i - t_i\}_1^n, \{R_i\}_1^n >$, where the rates R_i denote the entropy of the i^{th} source. For convenience, if all the sources are unit entropy, we will refer to the instance by just $< G, \{s_i - t_i\}_1^n >$, where the $s_i - t_i$ connections will occasionally be referred to as sessions that we need to support.

The instance is said to have a scalar linear network coding solution if there exist a set of linear encoding coefficients for each node in V such that each terminal t_i can recover X_i using the received symbols at its input edges. Likewise, it is said to have a vector linear network coding solution with vector length T if the network employs vector linear network codes and each terminal t_i can recover $[X_i^{(1)} \ \ldots \ X_i^{(T)}]$.

We will also be interested in examining the existence of a routing solution, wherever possible. In a routing solution, each edge carries a copy of one of the sources, i.e., each coding vector is such that at most one entry takes the value 1, all others are 0. Scalar (vector) routing solutions can be defined in a manner similar to scalar (vector) network codes. We now define some quantities that shall be used throughout the paper.

Definition 2.1: Connectivity level. The connectivity level for source-terminal pair $s_i - t_i$ is said to be n if the maximum flow between s_i and t_i in G is n. The connectivity level of the set of connections $s_1 - t_1, \ldots, s_n - t_n$ is the vector [max-flow($s_1 - t_1$) max-flow($s_2 - t_2$) $\ldots$ max-flow($s_n - t_n$)].

In this work our aim is to characterize the feasibility of the multiple unicast problem based on the connectivity level of the $s_i - t_i$ pairs. The questions that we seek to answer are of the following form.

Suppose that the connectivity level is $[k_1 \ k_2 \ \ldots \ k_n]$. Does any instance always have a linear (scalar or vector) network coding solution? If not, is it possible to demonstrate a counter-example, i.e, an instance of a graph G and $s_i - t_i$'s such that recovering X_i at t_i for all i is impossible under linear (or nonlinear) strategies?

In this paper, our achievability results will be constructive and based on linear network coding, whereas the counter-examples will hold under all possible strategies.

III. Multiple unicast under routing

We begin by providing a simple condition that guarantees the existence of a routing solution.

Theorem 3.1: Consider a multiple unicast instance with n $s_i - t_i$ pairs such that the connectivity level is $[n \ n \ \ldots \ n]$. There exists a vector routing solution for this instance.

Proof: Under vector routing over n time units, source s_i observes $[X_i^{(1)} \ \ldots \ X_i^{(n)}]$ symbols. Each edge e in the original graph is replaced by n parallel edges, $e^1, e^2, \ldots, e^n$. Let G_α represent the subgraph of this graph consisting of edges with superscript α. It is evident that max-flow($s_\alpha - t_\alpha$) = n over G_α. Thus, we transmit $X_\alpha^{(1)}, \ldots, X_\alpha^{(n)}$ over G_α using routing, for all $\alpha = 1, \ldots, n$. It is clear that this strategy satisfies the demands of all the terminals. ■

Note that in general, a network with the above connectivity level may not be able to support a scalar routing solution, an instance is shown in Figure 1. However, a scalar network coding solution exists for this example.

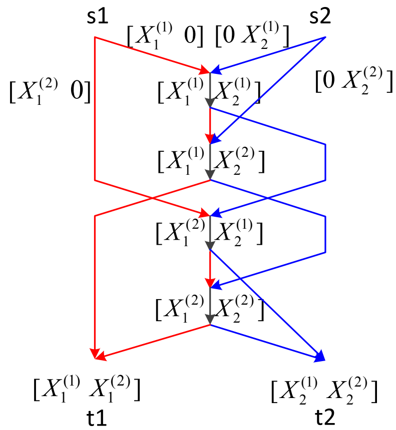

Fig. 1. A network with connectivity levels [2 2] and rate $\{1, 1\}$. There is a vector routing solution as shown in the figure. There is no scalar routing solution.

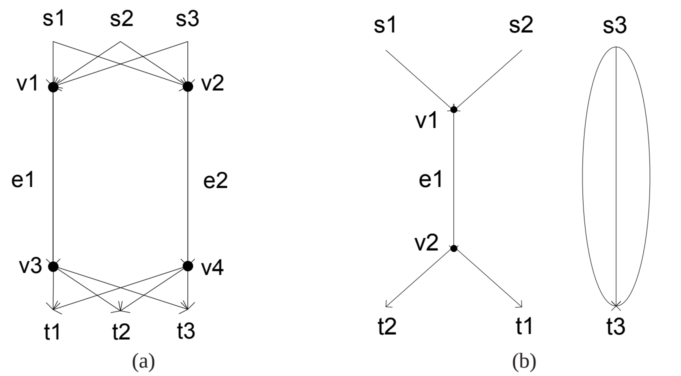

Fig. 2. (a) An example of [2 2 2] connectivity network without a network coding solution. (b) An example of [1 1 3] connectivity network without a network coding solution.

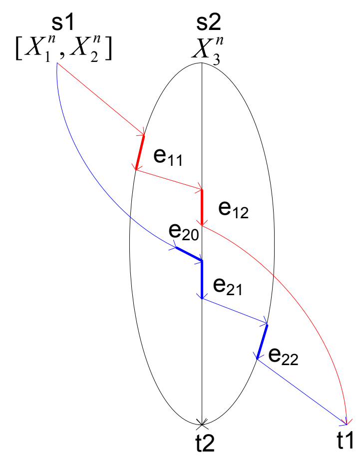

Fig. 3. An example of [2 3] connectivity network, rate $\{2, 1\}$ cannot be supported.

IV. Network coding for three unicast sessions

In the case of three unicast sessions, it is clear based on the results of Section III that if the connectivity level is [3 3 3], then a vector routing solution always exists. In this section we provide a full characterization of the feasibility/infeasibility of supporting three unicast sessions for a connectivity level of $[k_1\ k_2\ k_3]$, where $1 \leq k_i, \leq 3, i = 1, \ldots, 3$. For the feasible cases we will demonstrate appropriate linear network code assignments. On the other hand, for the infeasible cases we will present counter-examples where it is not possible to satisfy the terminal's demands under any coding strategy.

A. Infeasible Instances

We begin by demonstrating certain instances that can be ruled out by using cutset bounds.

Lemma 4.1: There exist multiple unicast instances with three unicast sessions such that the connectivity levels [2 2 2] and [1 1 3] are infeasible.

Proof: A network with connectivity levels [2 2 2] is shown in Figure 2(a). Consider the cut specified by the set of nodes $\{s_1, s_2, s_3, v_1, v_2\}$ that has a capacity value of 2. The rate that needs to be supported over $\{e_1, e_2\}$ is 3. By the cut set bound, this rate cannot be achieved.

Similarly, a network with connectivity levels [1 1 3] is shown in Figure 2(b). Consider the cut $\{s_1, s_2, v_1\}$. The capacity of this cut is 1. However, the rate that needs to be supported over e_1 is 2. Therefore, there does not exist a network coding solution. ■

While cut set bound is useful in the above cases, there exist certain connectivity levels for which a cut set bound is not tight enough. We now present such an instance in Figure 3. We show that this instance is not feasible under any code scheme (linear or nonlinear). This instance was also presented in the work of Erez and Feder [11], though they did not provide a formal proof of this fact.

Lemma 4.2: There exists a multiple unicast instance, with two sessions $< G, \{s_1 - t_1, s_2 - t_2\}, \{2, 1\} >$ and connectivity level [2 3] that is infeasible.

Proof: The graph instance is shown in Figure 3. Assume in n time units, s_1 observes two independent vector sources $[X_1^{(1)}\ \ldots\ X_1^{(n)}]$ and $[X_2^{(1)}\ \ldots\ X_2^{(n)}]$, s_2 observes one independent vector source $[X_3^{(1)}\ \ldots\ X_3^{(n)}]$. The sources are denoted as X_1^n, X_2^n and X_3^n for simplicity. The n random variables that e_i carries are denoted as $Y_{e_i}^n$, or simply Y_i^n. Suppose that the alphabet of X_i is $\mathcal{X}$. Since the entropy rates for the three sources are the same, we can assume $H(X_i) = \log|\mathcal{X}| = a$. Also, since we are interested in the feasibility of the solution, we can further assume that the alphabet size of Y_{ij} is also the same as $\mathcal{X}$, and $H(Y_{ij}) \leq \log|\mathcal{X}| = a$ by the capacity constraint of the edge. At terminal t_1 and t_2, from Y_{11}^n, Y_{12}^n, Y_{21}^n and Y_{22}^n, we estimate X_1^n, X_2^n and X_3^n. Let the estimate be $\widehat{X}_1^n$, $\widehat{X}_2^n$ and $\widehat{X}_3^n$. Suppose that there exist network codes and decoding function such that $P((\widehat{X}_1^n, \widehat{X}_2^n) \neq (X_1^n, X_2^n)) \to 0$ as $n \to \infty$. From the Fano's inequality, we shall have

$$H(X_1^n, X_2^n | \widehat{X}_1^n, \widehat{X}_2^n) \leq n\epsilon_n. \tag{1}$$

where $n\epsilon_n = 1 + nP_e \log(|\mathcal{X}|)$. For t_1 to decode X_1^n and X_2^n asymptotically, $\epsilon_n \to 0$ as $P_e \to 0$, when $n \to \infty$, where

$P_e = P((\widehat{X}_1^n, \widehat{X}_2^n) \neq (X_1^n, X_2^n))$.

Likewise, decodability at t_1 implies that $\widehat{X}_1^n, \widehat{X}_2^n$ are functions of Y_{12}^n and Y_{22}^n. Hence, we will have

$$\begin{aligned} H(X_1^n, X_2^n|Y_{12}^n, Y_{22}^n) &= H(X_1^n, X_2^n|\widehat{X}_1^n, \widehat{X}_2^n, Y_{12}^n, Y_{22}^n) \\ &\leq H(X_1^n, X_2^n|\widehat{X}_1^n, \widehat{X}_2^n) \leq n\epsilon_n. \end{aligned} \tag{2}$$

Now the sequences of information coming into t_1 are,

$$\begin{aligned} 2an &\overset{(a)}{\geq} H(Y_{12}^n, Y_{22}^n) \\ &\overset{(b)}{=} H(Y_{12}^n, Y_{22}^n, X_1^n, X_2^n) - H(X_1^n, X_2^n|Y_{12}^n, Y_{22}^n) \\ &\geq H(X_1^n, X_2^n) - H(X_1^n, X_2^n|Y_{12}^n, Y_{22}^n) \\ &\overset{(c)}{\geq} 2an - n\epsilon_n \end{aligned} \tag{3}$$

(a) is due to the capacity constraints of the edge e_{12} and e_{22}. (b) follows from the chain rule. (c) is because rate $2an$ should be transmitted over n time units and Equation (2).

Next, we shall have

$$\begin{aligned} &H(Y_{12}^n, Y_{22}^n|X_1^n, X_2^n) \\ &\overset{(a)}{=} H(Y_{12}^n, Y_{22}^n, X_1^n, X_2^n) - H(X_1^n, X_2^n) \\ &\overset{(b)}{=} H(X_1^n, X_2^n|Y_{12}^n, Y_{22}^n) + H(Y_{12}^n, Y_{22}^n) - H(X_1^n, X_2^n) \\ &\overset{(c)}{\leq} n\epsilon_n + 2an - 2an = n\epsilon_n. \end{aligned} \tag{4}$$

(a)(b) follows from the chain rule. (c) is from Equation (2) and Equation (3).

Analyzing the independence of X_1^n, X_2^n, and X_3^n, we shall have

$$\begin{aligned} an &= H(X_3^n|X_1^n, X_2^n) \\ &\overset{(a)}{=} H(X_3^n|Y_{12}^n, Y_{22}^n, X_1^n, X_2^n) + I(X_3^n; Y_{12}^n, Y_{22}^n|X_1^n, X_2^n) \\ &= H(X_3^n|Y_{12}^n, Y_{22}^n, X_1^n, X_2^n) + H(Y_{12}^n, Y_{22}^n|X_1^n, X_2^n) \\ &\quad - H(Y_{12}^n, Y_{22}^n|X_1^n, X_2^n, X_3^n) \\ &\overset{(b)}{\leq} H(X_3^n|Y_{12}^n, Y_{22}^n, X_1^n, X_2^n) + n\epsilon_n \\ &\overset{(c)}{\leq} H(X_3^n|Y_{12}^n, Y_{22}^n) + n\epsilon_n \overset{(d)}{\leq} an + n\epsilon_n \end{aligned} \tag{5}$$

(a) is from the definition of conditional mutual information. (b) is from Equation (4) and because conditioning reduces entropy. (c) is because conditioning reduces entropy. (d) is because conditioning reduces entropy. From the above inequalities, the information on e_{12} and e_{22} cannot decode X_3^n asymptotically. Then we have the following equations,

$$an - n\epsilon_n \leq H(X_3^n|Y_{12}^n, Y_{22}^n) \leq an \tag{6}$$

$$I(Y_{12}^n, Y_{22}^n; X_3^n) = H(X_3^n) - H(X_3^n|Y_{12}^n, Y_{22}^n) \leq n\epsilon_n; \tag{7}$$

$$\begin{aligned} H(Y_{12}^n, Y_{22}^n|X_3^n) &= H(Y_{12}^n, Y_{22}^n) - I(Y_{12}^n, Y_{22}^n; X_3^n) \\ &\geq 2an - 2n\epsilon_n \end{aligned} \tag{8}$$

$$I(Y_{12}^n; X_3^n) = I(Y_{12}^n, Y_{22}^n; X_3^n) - I(Y_{22}^n; X_3^n|Y_{12}^n) \leq n\epsilon_n;$$

$$I(Y_{22}^n; X_3^n) \leq n\epsilon_n \tag{9}$$

The above inequalities imply that the information on e_{12} and e_{22} are asymptotically independent of X_3^n.

Because Y_{21}^n is only a function of Y_{12}^n and Y_{20}^n,

$$\begin{aligned} &H(Y_{21}^n, Y_{22}^n) \\ &\overset{(a)}{=} H(X_3^n, Y_{21}^n, Y_{22}^n) - H(X_3^n|Y_{21}^n, Y_{22}^n) \\ &\overset{(b)}{=} H(X_3^n, Y_{21}^n) - H(X_3^n|Y_{21}^n, Y_{22}^n) \\ &\overset{(c)}{\leq} 2an - H(X_3^n|Y_{21}^n, Y_{22}^n) \\ &\overset{(d)}{\leq} 2an - H(X_3^n|Y_{21}^n, Y_{22}^n, Y_{20}^n, Y_{12}^n, X_1^n, X_2^n) \\ &\overset{(e)}{=} 2an - H(X_3^n|Y_{22}^n, Y_{20}^n, Y_{12}^n, X_1^n, X_2^n) \\ &\overset{(f)}{=} 2an - H(X_3^n|Y_{22}^n, X_1^n, X_2^n, Y_{12}^n) \\ &\overset{(g)}{=} 2an - H(X_3^n|Y_{22}^n, Y_{12}^n) + I(X_3^n; X_1^n, X_2^n|Y_{22}^n, Y_{12}^n) \\ &\overset{(h)}{=} 2an - H(X_3^n|Y_{22}^n, Y_{12}^n) + H(X_1^n, X_2^n|Y_{22}^n, Y_{12}^n) \\ &\qquad - H(X_1^n, X_2^n|Y_{22}^n, X_3^n, Y_{12}^n) \\ &\leq 2an - H(X_3^n|Y_{22}^n, Y_{12}^n) + H(X_1^n, X_2^n|Y_{22}^n, Y_{12}^n) \\ &\overset{(i)}{\leq} 2an - an + n\epsilon_n + n\epsilon_n = an + 2n\epsilon_n \end{aligned} \tag{10}$$

(a) follows from the chain rule, (b) is because Y_{22}^n is a function of X_3^n and Y_{21}^n. (c) is because of the capacity constraints. (d) is because conditioning reduces entropy. (e) is because Y_{21}^n is a function of Y_{12}^n and Y_{20}^n. (f) is because Y_{20}^n is a function of X_1^n and X_2^n. (g)(h) follows from the mutual information definition. (i) is from Equation (2) and Equation (6). The above inequalities indicate that e_{21} and e_{22} should carry the same information asymptotically.

From the network, we know that Y_{12}^n is a function of Y_{11}^n and X_3^n. Then

$$\begin{aligned} H(Y_{11}^n, Y_{21}^n, Y_{22}^n|X_3^n) &= H(Y_{11}^n, Y_{21}^n, Y_{22}^n, X_3^n|X_3^n) \\ &\geq H(Y_{12}^n, Y_{21}^n, Y_{22}^n|X_3^n) \\ &\geq H(Y_{22}^n, Y_{12}^n|X_3^n) \overset{(a)}{\geq} 2an - 2n\epsilon_n \end{aligned} \tag{11}$$

(a) is due to Equation (8).

Finally, we shall have

$$\begin{aligned} &H(X_3^n|Y_{11}^n, Y_{21}^n, Y_{22}^n) \\ &= H(Y_{11}^n, Y_{21}^n, Y_{22}^n|X_3^n) + H(X_3^n) - H(Y_{22}^n, Y_{21}^n, Y_{11}^n) \\ &\overset{(a)}{\geq} 2an - 2n\epsilon_n + an - H(Y_{22}^n, Y_{21}^n, Y_{11}^n) \\ &= 3an - 2n\epsilon_n - H(Y_{22}^n, Y_{21}^n) - H(Y_{11}^n|Y_{22}^n, Y_{21}^n) \\ &\overset{(b)}{\geq} 3an - 2n\epsilon_n - an - 2n\epsilon_n - H(Y_{11}^n|Y_{22}^n, Y_{21}^n) \\ &\overset{(c)}{\geq} 2an - 4n\epsilon_n - an = an - 4n\epsilon_n \end{aligned} \tag{12}$$

(a) is because of Equation (11). (b) is because of Equation (10). (c) is due to the capacity constraint of Y_{11}^n.

When $n \to \infty$, for t_1 to asymptotically decode X_1^n and X_2^n, we shall have $\epsilon_n \to 0$. Then t_2 cannot decode X_3^n asymptotically. ∎

Corollary 4.3: There exists a multiple unicast instance with three sessions, and connectivity level $[2\ 3\ 2]$ that is infeasible.

Proof: Consider a multiple unicast instance $< G, \{s_i' - t_i'\}_1^3, \{1, 1, 1\} >$, where G is the graph in Figure 3. The sources s_1' and s_3' are collocated at s_1 (in G), and the terminals t_1' and t_3' are collocated at t_1 (in G). Likewise, the source s_2' and

terminal t_2' are located at s_2 and t_2 in G. The three sessions have connectivity level $[2\ 3\ 2]$. Based on the arguments in Lemma 4.2, there is no feasible solution for this instance. ■

The instance presented in Lemma 4.2, can be generalized to obtain a series of counter-examples. In particular, we have the following theorem shows an instance with two unicast sessions with connectivity level $[n_1\ n_2]$ that cannot support rates $R_1 = n_1, R_2 = n_2 - n_1$.

Theorem 4.4: For a directed acyclic graph G with two $s-t$ pairs, if the connectivity level for (s_1, t_1) is n_1, for (s_2, t_2) is n_2, $1 < n_1 < n_2$, there exist instances that cannot support $R_1 = n_1$ and $R_2 = n_2 - n_1$.

Proof: The proof is omitted due to space limitations. ■

B. Feasible Instances

It is evident that the infeasibility of an instance with connectivity level $[2\ 2\ 3]$ implies that when $1 \leq k_i \leq 3$, the only possible instances that are potentially feasible are [1 3 3], its permutations and connectivity levels that are greater than it. We now show that many of these instances are feasible using linear network codes. In this subsection, we present efficient linear network code assignment algorithms for these cases. Towards this end, we need the following definitions.

Definition 4.5: Minimality. Consider a multiple unicast instance $< G = (V, E), \{s_i - t_i\}_1^n >$, with connectivity level $[k_1\ k_2\ \ldots\ k_n]$. The graph G is said to be minimal if the removal of any edge from E strictly reduces the connectivity level. If G is minimal, we will also refer to the multiple unicast instance as minimal.

Clearly, given a non-minimal instance $G = (V, E)$, we can always remove the non-essential edges from it, to obtain the minimal graph $G_{\min}$. This does not affect feasibility, since a network code for $G_{\min} = (V, E_{\min})$ can be converted into a network code for G by simply assigning the all-zeros coding vector to the edges in $E \backslash E_{\min}$.

Definition 4.6: Overlap edge. An edge e is said to be an overlap edge for paths P_i and P_j in G, if $e \in P_i \cap P_j$.

Definition 4.7: Overlap segment. In G, consider an ordered set of edges $E_{os} = \{e_1, \ldots, e_l\} \subset E$ that forms a path. This path is called an overlap segment for paths P_i and P_j if

(i) $\forall k \in \{1, \ldots, l\}$, e_k is an overlap edge for P_i and P_j.

(ii) None of the incoming edges into tail(e_1) are overlap edges for P_i and P_j.

(iii) None of the outgoing edges leaving head(e_l) are overlap edges for P_i and P_j.

Our solution strategy is as follows. We first convert the original instance into another *structured* instance where each internal node has at most degree three (in-degree + out-degree). We then convert this new instance into a minimal one, and then develop the code assignment algorithm. It will be evident that using this network code, one can obtain a network code for the original instance.

1) Conversion procedure: Let $G = (V, E)$ be our original graph, and let s_i and t_i be the given sources and terminals. We can efficiently construct a *structured* graph $\hat{G} = (\hat{V}, \hat{E})$ in which each internal node $v \in \hat{V}$ is of total degree at most three with the additional following properties: (a) $\hat{G}$ is acyclic. (b) For every source (terminal) in G there is a corresponding source (terminal) in $\hat{G}$. (c) For any two edge disjoint paths P_i and P_j for one unicast session in G, there exist two *vertex* disjoint paths in $\hat{G}$ for the corresponding session in $\hat{G}$. (d) Any feasible network coding solution in $\hat{G}$ can be efficiently turned into a feasible network coding solution in G. Our reduction steps are the same as in [12]. Due to space limitations, refer to [12] and [13] for details.

2) Code Assignment Procedure: In the discussion below, we will assume that the graph G is structured. It is clear that this is without loss of generality based on the previous arguments. In our arguments, we will use the minimality of the graph extensively.

Lemma 4.8: Consider a minimal multiple unicast instance, $< G, \{s_1 - t_1, s_2 - t_2\} >$ with connectivity level $[1\ m]$. Denote the $s_1 - t_1$ path by P_1 and the set of edge disjoint $s_2 - t_2$ paths by $\{P_{21}, \ldots, P_{2m}\}$. There can be at most one overlap segment between P_1 and each $P_{2i}, i = 1, \ldots, m$.

Proof: Suppose that there are two overlap segments $E_{os1} = \{e_1, \ldots, e_{k_1}\}$ and $E_{os2} = \{e_1', \ldots, e_{k_2}'\}$ between P_1 and P_{2i}, where e_{k_1} is upstream of e_1'. Note that by the definition of an overlap segment and the fact that G is structured, it holds that the head of e_{k_1} has in-degree one and out-degree two, so that one outgoing edge from $head(e_{k_1})$ belongs to P_1 (denoted e^*) and the other belongs to P_{2i}. Note also $e^* \in P_1$ cannot belong to $P_{2j}, j \neq i$ since the set of paths $\{P_{21}, \ldots, P_{2m}\}$ is vertex disjoint (since G is structured).

Now, note that e^* can be removed while still maintaining the required connectivity level. This is true for $s_2 - t_2$, since e^* does not lie on any of the paths $P_{21}, \ldots, P_{2m}$. It is true for $s_1 - t_1$ since there is a path from e_{k_1} to e_{k_2}' that overlaps P_{2i}, and therefore this still continues be a path from $s_1 - t_1$. This path can be explicitly specified as $path(s_1, head(e_{k_1})), path(e_{k_1}, e_{k_2}'), path(head(e_{k_2}'), t_1)$. ■

Using this property, we can obtain the following result that holds for the case of two unicast sessions with the rate $\{1, m\}$.

Lemma 4.9: A minimal multiple unicast instance $< G, \{s_1 - t_1, s_2 - t_2\}, \{1, m\} >$ with connectivity level $[1\ m+1]$ is always feasible.

Proof: We show that this can be achieved by using scalar linear network codes. Let P_1 denote the path from $s_1 - t_1$ and $m + 1$ vertex-disjoint paths from $s_2 - t_2$, as $P_{2j}, j = 1, \ldots, m+1$. Let the source message at s_1 be denoted by X_1 and the source message vector at s_2 by $[X_{21}, \ldots, X_{2m}]$. We proceed by induction on m.

Base case - $m = 1$. In this case suppose that P_1 intersects at most one path from the $s_2 - t_2$. For instance if P_1 overlaps with P_{21}, then simply transmit X_{21} over P_{22} and X_1 over P_1.

Alternatively, P_1 overlaps both P_{21} and P_{22}. Suppose that the segments are denoted E_{os1} and E_{os2} respectively and that E_{os1} is upstream of E_{os2} (w.l.o.g.). In this case, we flow X_1 (X_{21}) on P_1 (P_{21}) until E_{os1} and flow $X_1 + X_{21}$ on E_{os1}, and further downstream on P_{21} till t_2 and on P_1 until E_{os2}. We flow X_{21} on P_{22} until E_{os2} and flow $X_1 + X_{21} + X_{21} = X_1$,

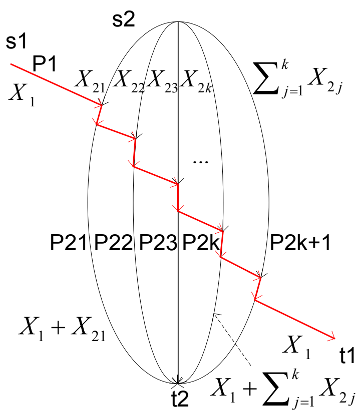

Fig. 4. An example where P_1 overlaps with all paths $P_{21}, \ldots, P_{2k+1}$. Rate $\{1, k\}$ is feasible.

on E_{os2} and further downstream till t_1 and t_2. It is evident that t_2 can recover X_{21} from its received values.

Induction step. Suppose that the induction hypothesis holds for $m = k$. For $m = k + 1$, again we consider two cases. Suppose that P_1 does not overlap with at least one path from the set $\{P_{21}, \ldots, P_{2k+1}\}$, w.l.o.g. suppose that it is P_{2k+1}. In this case the graph consisting of $P_1 \cup P_{21} \cup \cdots \cup P_{2k}$, can be used to transmit X_1 to t_1 and $X_{21}, \ldots, X_{2k-1}$ to t_2 using the induction hypothesis. X_{2k} can simply be routed on P_{2k+1}.

On the other hand if P_1 overlaps with all the paths $P_{21}, \ldots, P_{2k+1}$. We assume w.l.o.g. that it overlaps first with P_{21} (in E_{os1}), then with P_{22} and so on until P_{2k+1}. In this case, as illustrated in Figure 4, we can arrive at the required solution. In particular, s_2 transmits X_{2i} over paths $P_{2i}, i = 1, \ldots, k$ and $\sum_{j=1}^{k} X_{2j}$ over P_{2k+1} until the overlap point. The path P_1 carries X_1 until E_{os1}. At each overlap segment a sum of the incoming values into the segment is computed. This ensures that overlap segment E_{osi} carries $X_1 + \sum_{j=1}^{i} X_{2j}, i = 1, \ldots, k$ and E_{osk+1} carries X_1. It can be seen that both t_1 and t_2 are satisfied in this case. ∎

It turns out that one can treat the case of three multiple unicast sessions with connectivity level $[1\ 3\ 3]$, by using the result of Lemma 4.9. The basic idea is to use vector linear network coding over two time units and code over pairs of sources at appropriately defined layers of the network. We state and prove this result below.

Theorem 4.10: A multiple unicast instance with three sessions such that the connectivity level is $[1\ 3\ 3]$ is always feasible.

Proof: Let the original graph (with unit capacity edges) be denoted by $G = (V, E)$. We use vector linear network coding over two time units, i.e. $T = 2$. In this case we form a new graph G^* where each edge $e \in E$ is replaced by two parallel unit capacity edges e^1 and e^2 in G^*. The messages at source node s_i are denoted $[X_{i1}\ X_{i2}]$. Now consider the subgraph of G^* induced by all edges with superscript 1, that we denote G_1^*. In G_1^*, there exists a single $s_1 - t_1$ path and three edge disjoint $s_2 - t_2$ paths. Therefore, we can transmit X_{11} from s_1 to t_1 and $[X_{21}\ X_{22}]$ from $s_2 - t_2$ using the result of Lemma 4.9. Similarly, we use the subgraph of G^* induced by all edges with superscript 2, i.e., G_2^* to communicate X_{12} from s_1 to t_1 and $[X_{31}\ X_{32}]$ from s_3 to t_3. Thus, using vector linear network coding over two time units, a connectivity level of $[1\ 3\ 3]$ suffices to satisfy the demands of each terminal. ∎

Corollary 4.11: A multiple unicast instance with three sessions such that the connectivity level is greater than $[1\ 3\ 3]$ is always feasible.

Proof: For the graph G which has connectivity level greater than $[1\ 3\ 3]$, we identify a subgraph G' with connectivity level $[1\ 3\ 3]$. By Theorem 4.10, the demand at each terminal can be satisfied. Then by assigning zero coding vector to the edges in $G \setminus G'$, the terminal demand can be satisfied in the original graph G. ∎

So far, we have completely characterized the cases where the connectivity levels are $[k_1\ k_2\ k_3]$, $k_i \leq 3$. However, there are several connectivity levels with unknown feasibility when $k_i > 3$, e.g., $[2\ 2\ 4]$.

V. Conclusions and Future Work

In this work, we have identified several feasible and infeasible connectivity levels for 3 unicast sessions. For the feasible instances, we provided explicit network code assignments, while for the infeasible instances we demonstrated appropriate counter-examples. Some of these results can be extended to the case of general n, and are currently under investigation.

References

[1] R. Koetter and M. Médard, "An Algebraic approach to network coding," *IEEE/ACM Trans. on Netw.*, vol. 11, no. 5, pp. 782–795, 2003.
[2] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network Information Flow," *IEEE Trans. on Info. Th.*, vol. 46, no. 4, pp. 1204–1216, 2000.
[3] D. Traskov, N. Ratnakar, D. Lun, R. Koetter, and M. Medard, "Network Coding for Multiple Unicasts: An Approach based on Linear Optimization," in *IEEE Intl. Symposium on Info. Th.*, 2006, pp. 1758–1762.
[4] C.-C. Wang and N. B. Shroff, "Pairwise Intersession Network Coding on Directed Networks," *IEEE Trans. on Info. Th.*, vol. 56, no. 8, pp. 3879–3900, Aug, 2010.
[5] N. Harvey, R. Kleinberg, and A. Lehman, "On the capacity of information networks," *IEEE Trans. on Info. Th.*, vol. 52, no. 6, pp. 2345–2364, 2006.
[6] Z. Li and B. Li, "Network coding: the Case of Multiple Unicast Sessions," in *42st Allerton Conference on Communication, Control, and Computing*, 2004.
[7] X. Yan, R. W. Yeung, and Z. Zhang, "The Capacity Region for Multi-source Multi-sink Network Coding," in *IEEE Intl. Symposium on Info. Th.*, June, 2007, pp. 116–120.
[8] A. Das, S. Vishwanath, S. A. Jafar, and A. Markopoulou, "Network Coding for Multiple Unicasts: An Interference Alignment Approach," in *IEEE Intl. Symposium on Info. Th.*, 2010, pp. 1878 – 1882.
[9] A. E. Kamal, A. Ramamoorthy, L. Long, and S. Li, "Overlay protection against link failures using network coding," *IEEE/ACM Trans. on Netw.*, (to appear), 2011.
[10] S. Li and A. Ramamoorthy, "Protection against link errors and failures using network coding in overlay networks," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, July 2009, pp. 986 –990.
[11] E. Erez and M. Feder, "Improving the Multicommodity Flow Rate with Network Codes for Two Sources," *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, vol. 27, no. 5, pp. 814–824, 2009.
[12] M. Langberg, A. Sprintson, and J. Bruck, "The encoding complexity of network coding," *IEEE Trans. on Info. Th.*, vol. 52, no. 6, pp. 2368–2397, 2006.

[13] A. Ramamoorthy and M. Langberg, “Communicating the sum of sources over a network,” *Submitted. (available at http://www.arxiv.org/abs/1001.5319)*.